
Manual de compilación de la librería ARUCO3 con OpenCV4 en Windows

---

http://umh1782.edu.umh.es

Requisitos: *Windows 7, Windows 10 - (64bits)*

Más información en http://www.uco.es/investiga/grupos/ava/node/26

1) Instalar Microsoft Visual Studio 2017 y Cmake:

- En el blog de la asignatura (Apartado ***OpenCV***) disponemos de un enlace para descargar el instalador tanto de Visual Studio 2017 como de CMake

  http://umh1782.edu.umh.es/opencv/#Software

- Al arrancar el compilador por primera vez, el programa nos pedirá una licencia que podemos conseguir de forma gratuita registrándonos en la web de Microsoft, basta seguir el enlace que nos indica.

3) Descargar la librería ARUCO:

- ARUCO es una librería OpenSource con licencia BSD desarrollada por el grupo AVA de la Universidad de Córdoba para la implementación de aplicaciones de Realidad Aumentada mediante la detección de marcadores codificados. Está basada en la libraría OpenCV, por lo que debemos tener esta librería instalada en nuestro ordenador (Manual de instalación de OpenCV)

- Podemos descargar el código fuente de la librería desde la web del proyecto ARUCO (http://sourceforge.net/projects/aruco/), la versión actual es la 3.1.12

- Descargaremos la última versión (actualmente aruco-3.1.12.tgz):

  https://sourceforge.net/projects/aruco/files/3.1.12/aruco-3.1.12.zip

Descomprimiremos el fichero y crearemos una carpeta con nombre '***build***' dentro de la carpeta de código fuente de ARUCO. Alternativamente podemos ejecutar los siguientes comandos en la consola:

```
cd aruco-3.1.12
mkdir build
cd build
```

4) Compilar la librería ARUCO:

4.1) Generar proyecto en Visual Studio 2017:

Ejecutaremos el programa ***CMake-Gui***:

- Seleccionaremos la ubicación de la carpeta del código de la librería (***aruco-3.1.12***)

- Seleccionaremos la carpeta donde estará ubicada la librería compilada. Crearemos una carpeta denominada (***build***) ***(***Figura 1***)***.

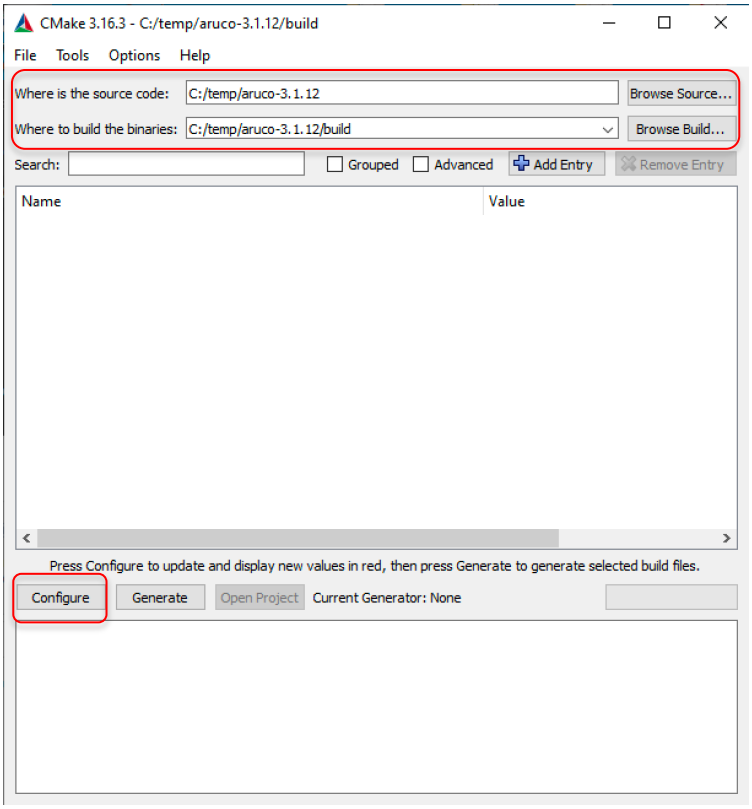

Figura 1 Cmake-gui

- Pulsaremos el botón "Configure" par a configurar el proyecto. Nos abrirá una ventana de configuración (**Figura 2**), donde seleccionaremos el compilador a utilizar, en nuestro caso **Visual Studio 2017 (VC15)**, así como la plataforma opcional **x64**.

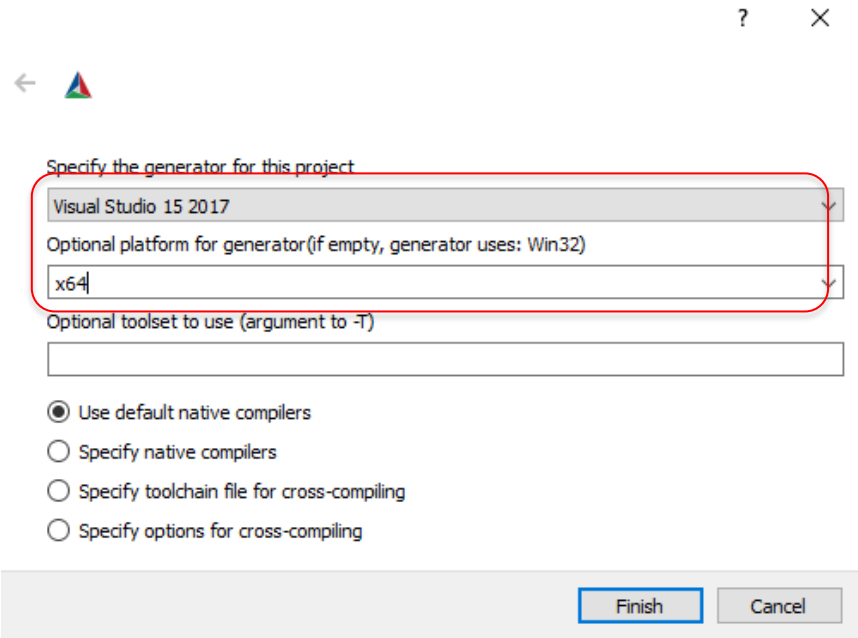

Figura 2. Configuración Compilador

- Al pulsar el botón "***Finish***" CMake iniciará la búsqueda en el código fuente de ARUCO y del fichero "***CMakeLists.txt***" y chequeará la configuración del compilador y otras herramientas necesarias. En la ventana (**Figura 3**) nos mostrará en la parte superior las opciones configurables en la compilación de OpenCV (se marcan en rojo las opciones añadidas desde la configuración previa, no se trata de errores). En la parte inferior nos mostrará avisos de la ejecución con posibles problemas (**texto en rojo**). Deberemos resolver cualquier aviso que nos dé. Cada vez que hagamos una modificación podemos testearla pulsando de nuevo el botón "***Configure***".

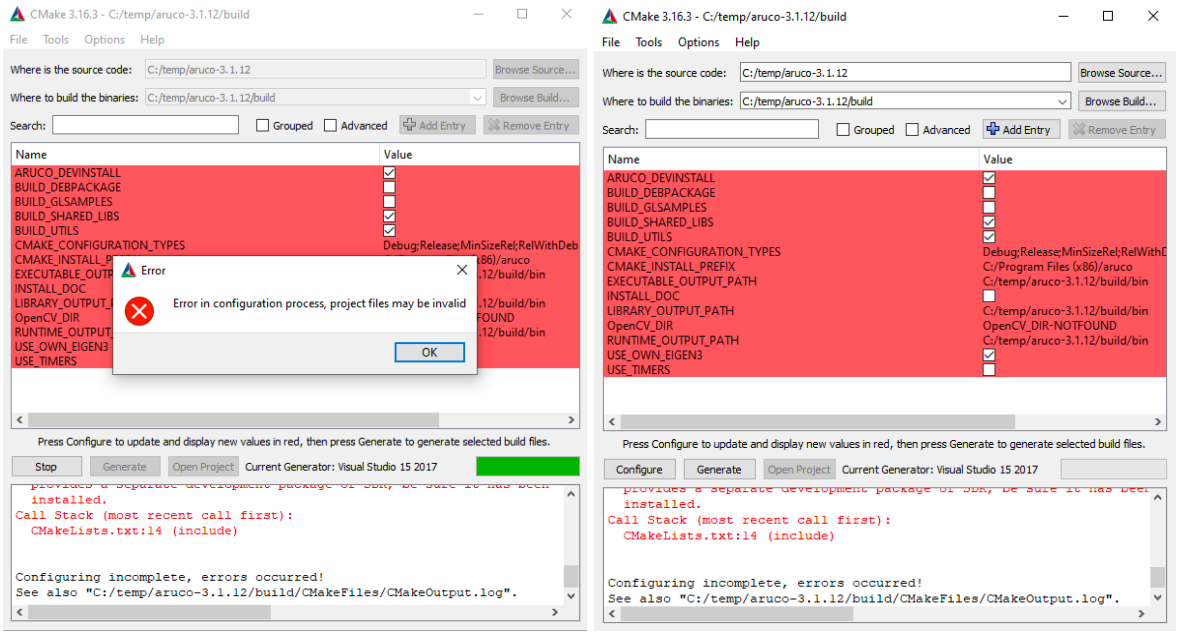

Figura 3. Resultado ejecución Cmake

En la (**Figura 3**) podemos ver que nos indica que hay un error, (NO ha conseguido localizar la librería **OpenCV** que precisa ARUCO: **OpenCV_DIR-NOTFOUND**

Editaremos la variable con la ubicación de la instalación de OpenCV: ***OpenCV_DIR = c:/opencv/build***. Volveremos a pulsar el botón 'Configure', podemos ver (Figura 4) que solo queda en rojo la opción GLUT, una librería de dibujo 3D que no necesitamos.

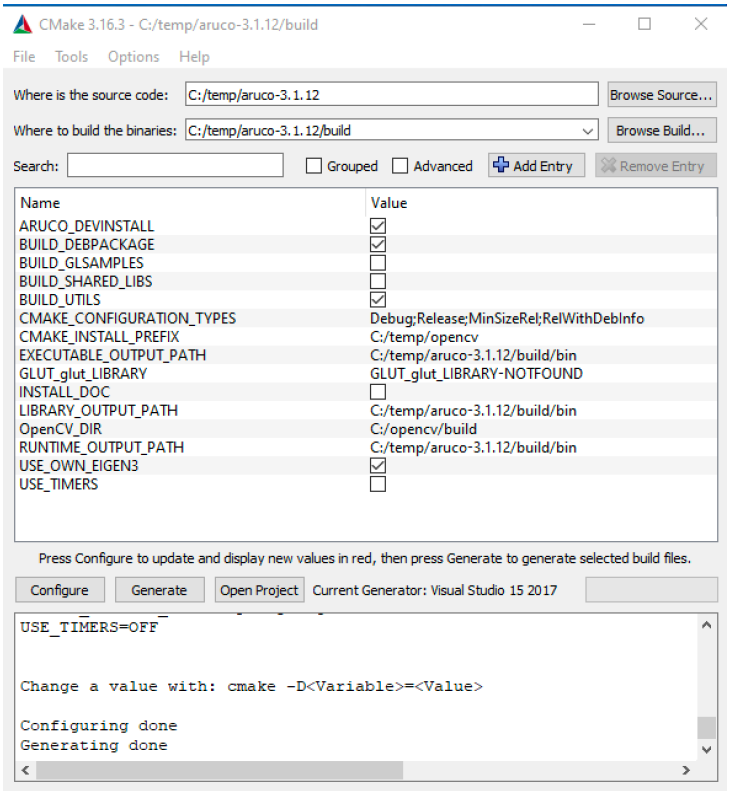

Figura 4. Configuración módulos adicionales (contrib)

- Configuración de otras opciones de OpenCV: en la ventana superior dispones de varias casillas con opciones que podemos activar o desactivar (están por orden alfabético). En nuestro caso configuraremos las siguientes opciones:

    - **CMAKE_INSTALL_PREFIX**: c:/temp/opencv
    - **BUILD_UTILS** (Activo)
    - **BUILD_SHARED_LIBS** (Desactivado) : crear una librería estática que no precisa de DLLs)

- Deberemos ir resolviendo cualquier aviso en rojo que nos aparezca, generalmente es que no tenemos instalado o no es capaz de encontrar algún paquete o librería. En internet podemos buscar la forma de resolverlo. Al finalizar volveremos a pulsar el botón ***Configure*** para comprobar que la configuración es correcta.

- A continuación (cuando hayamos resulto todos los errores) pulsaremos una última vez el botón "***Configure***" y a continuación el botón "***Generate***" para que genere los ficheros del proyecto en Visual Studio. Esto puede tardar algún tiempo.

  Nota: si quisiéramos empezar la configuración de CMake de cero, iremos al menú ***File*** y seleccionaremos la opción "***Delete Cache***".

4.2) Compilar el proyecto en Visual Studio 2017 (Linea de comandos):

Ejecutaremos los siguientes comandos en una consola con la carpeta de instalación seleccionada: ***c:/temp/aruco-3.1.12/build***

```
cmake --build . --config Debug --target INSTALL

rename c:\temp\opencv\lib\aruco3112.lib   aruco3112d.lib

cmake --build . --config Release --target INSTALL
```

Si hemos seleccionado la opción en CMake **CMAKE_INSTALL_PREFIX**: c:/temp/opencv

Se habrá creado la carpeta '***c:/temp/opencv***'. Hemos seleccionado esta carpeta para mezclar las librerías Aruco con las de OpenCV cuando completemos la instalación

Bastará copiar los contenidos de esta carpeta sobre la carpeta de instalación de OpenCV (*c:\opencv*)

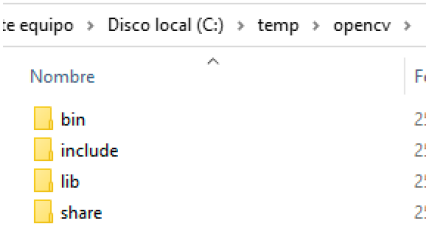

Figura 5. Directorio "install"

5) Configurar Hojas de propiedades compilación Visual Studio 2017:

Añadiremos a las hojas de propiedades configuradas para OpenCV la librería **Aruco**:

Editaremos la opción ***Vinculador -> Entrada -> Dependencias Adicionales***, pulsando sobre la flecha azul del lateral derecho nos muestra un menú con la opción <***Editar***> que pulsaremos y abrirá una ventana de edición. Añadiremos el listado de librerías ***Aruco*** tanto para la configuración *Release* como *Debug*.

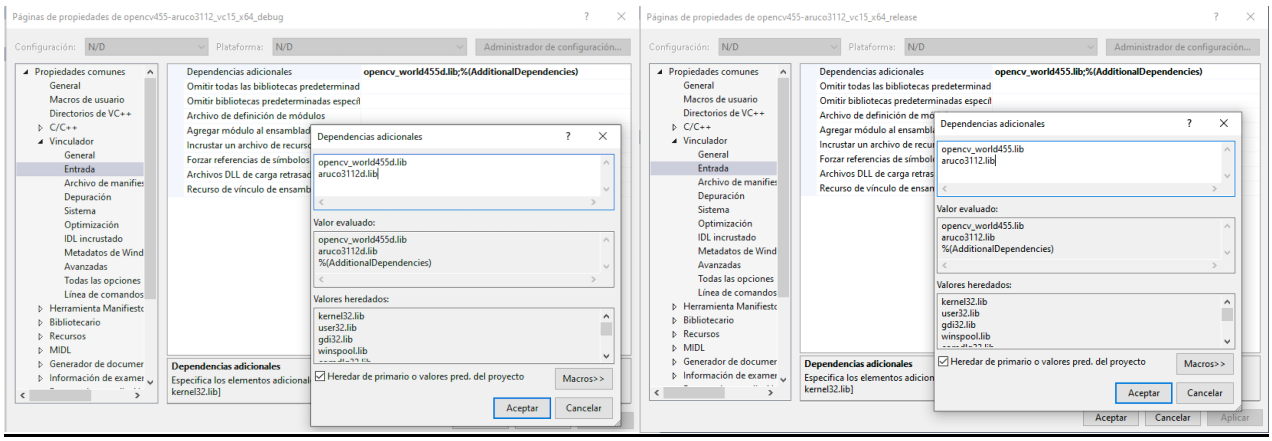

Figura 6. Configuración Hojas de Propiedades

Anexo- Compilar el proyecto en Visual Studio 2017 (Visual Studio):

Alternativamente podemos hacer la compilación desde Visual Studio 2017. Buscaremos la carpeta con el proyecto generado (***c:/temp/aruco-3.1.12/build***) y abriremos el fichero (***aruco.sln***). Al hacer doble click, se ejecutará el compilador y cargará el proyecto (puede tardar mientras analiza todos los ficheros del mismo, en el pie de la ventana podemos ver el progreso).

Nota: NO debemos cerrar la aplicación **CMake-Gui** por si debemos modificar alguna opción en caso de tener problemas con la compilación.

En la ventana izquierda del compilador (Figura 7) marcaremos la pestaña "*Explorador de Soluciones*". Podemos ver la configuración que ha realizado CMake para generar la compilación de cada parte de la librería y los ejemplos.

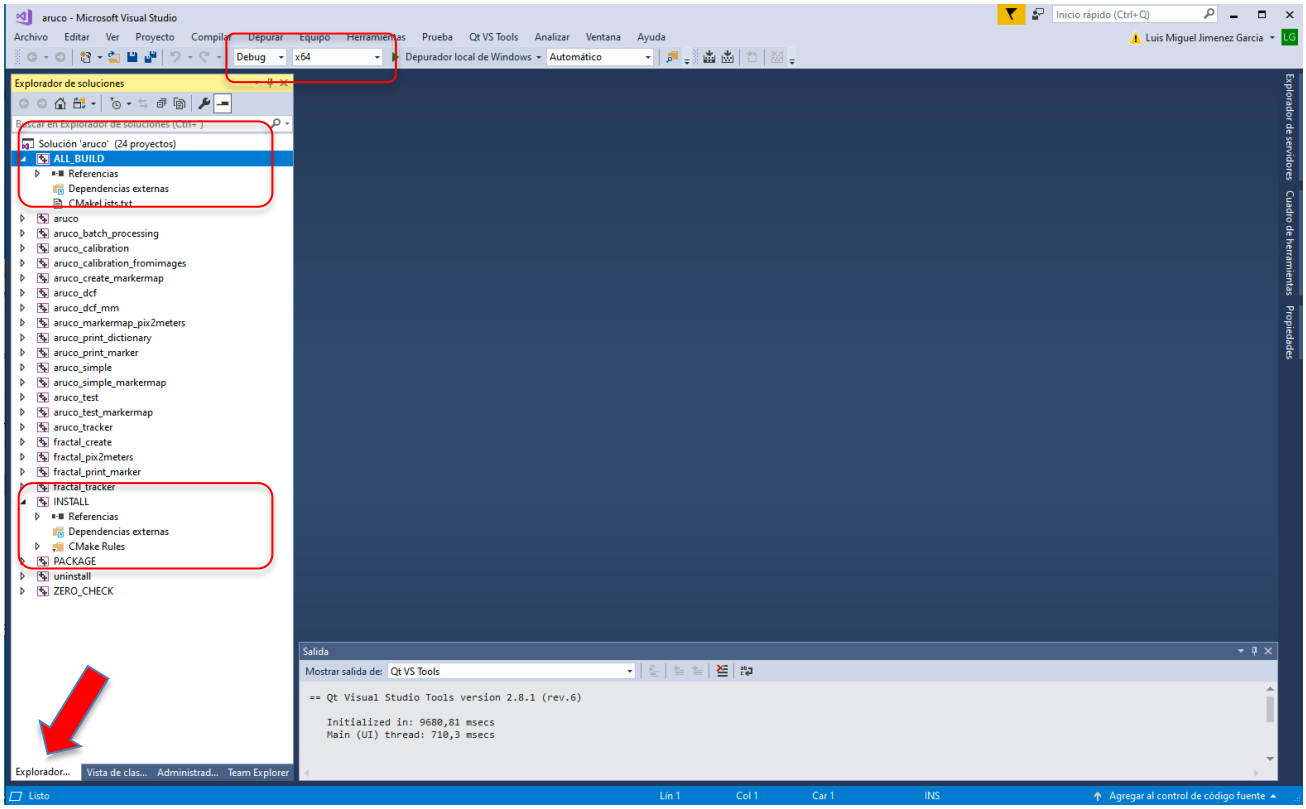

Figura 7. Proyecto librería Aruco en Visual studio 2017

Buscaremos la carpeta ***ALL_BUILD*** que nos permitirá hacer una compilación completa:

a) Compilar el elemento **ALL_BUILD (Debug-x64)**: en la pestaña superior seleccionares el modo ***Debug*** y la plataforma ***x64***. Seleccionaremos **ALL_BUILD** click derecho – opción compilar (Figura 8) ó seleccionar "***Compilar ALL_BUILD"*** desde menú ***Compilar***. En la ventana central (Salida) se muestra el avance de la compilación que puede tardar bastante tiempo.

b) Compilar el elemento **ALL_BUILD (Release-x64)**: en la pestaña superior seleccionares el modo ***Release*** y la plataforma ***x64***. Seleccionaremos **ALL_BUILD** click derecho – opción compilar (Figura 8) ó seleccionar "***Compilar ALL_BUILD***" desde menú ***Compilar***.

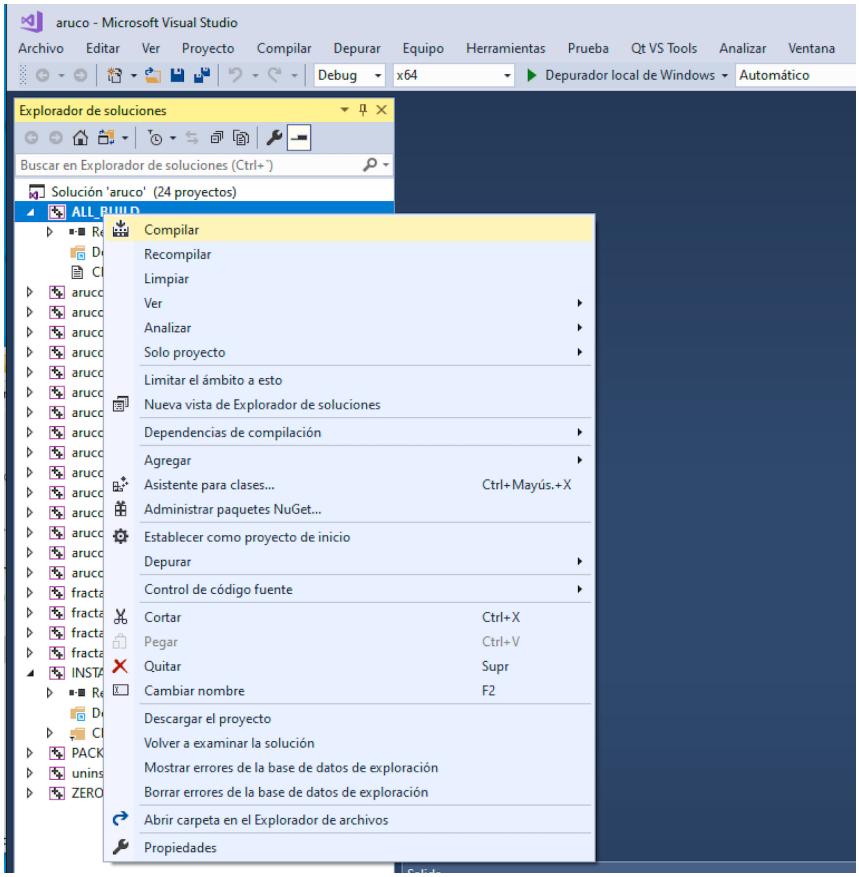

Figura 8. Compilación ALL_BUILD

c) Compilar el elemento **INSTALL (Debug-x64)**: en la pestaña superior seleccionares el modo ***Debug*** y la plataforma ***x64***. Seleccionaremos **INSTALL** click derecho – opción compilar (Figura 8) ó seleccionar "***Compilar* INSTALL"** desde menú ***Compilar***.

d) Compilar el elemento **INSTALL (Release-x64)**: en la pestaña superior seleccionares el modo ***Debug*** y la plataforma ***x64***. Seleccionaremos **INSTALL** click derecho – opción compilar (Figura 8) ó seleccionar "***Compilar* INSTALL"** desde menú ***Compilar***.

En la carpeta (***c:/temp/opencv***) (Figura 9) disponemos de los ficheros de la librería con toda su estructura de directorios. Podemos copiarlas en la carpeta de instalación estándar de OpenCV (***c:/opencv***)

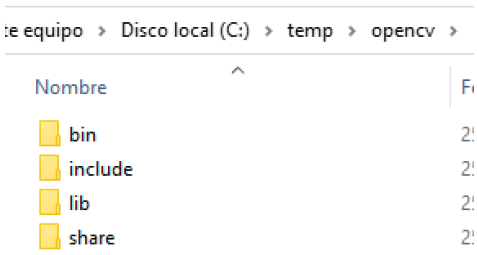

Figura 9. Directorio "install"